Biostat 200C Homework 1

Due Apr 16 @ 11:59PM

Zian ZHUANG

Q1. Binomial Distribution

Let Y_i be the number of successes in n_i trials with

$$Y_i \sim Bin(n_i, \pi_i),$$

where the probabilities π_i have a Beta distribution

$$\pi_i \sim Beta(\alpha, \beta).$$

The probability density function for the Beta distribution is $f(x; \alpha, \beta) = x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha, \beta)$ for $x \in [0, 1], \alpha > 0, \beta > 0$, and the beta function $B(\alpha, \beta)$ defining the normalizing constant required to ensure that $\int_0^1 f(x; \alpha, \beta) = 1$. Let $\theta = \alpha/(\alpha + \beta)$, show that

a. $E(\pi_i) = \theta$

$$\begin{aligned}
E(\pi_i) &= \int \pi_i * f(\pi_i) d\pi_i \\
&= \int \pi_i * \pi_i^{\alpha-1}(1-\pi_i)^{\beta-1}/B(\alpha, \beta) d\pi_i \\
&= B(\alpha, \beta)^{-1} \int \pi_i^{(\alpha+1)-1}(1-\pi_i)^{\beta-1} d\pi_i \\
&= B(\alpha+1, \beta) * B(\alpha, \beta)^{-1} \int B(\alpha+1, \beta)^{-1} * \pi_i^{(\alpha+1)-1}(1-\pi_i)^{\beta-1} d\pi_i \\
&= B(\alpha+1, \beta) * B(\alpha, \beta)^{-1} * 1 \\
&= \Gamma(\alpha+1)\Gamma(\beta)/\Gamma(\alpha+1+\beta)/(\Gamma(\alpha)\Gamma(\beta)) * \Gamma(\alpha+\beta) \\
&= \alpha/(\alpha+\beta) \\
&= \theta
\end{aligned}$$

b. $Var(\pi_i) = \theta(1-\theta)/(\alpha+\beta+1) = \phi\theta(1-\theta)$ Firstly we can calculated $E(\pi_i^2)$

$$\begin{aligned}
E(\pi_i^2) &= \int \pi_i^2 * f(\pi_i) d\pi_i \\
&= \int \pi_i^2 * \pi_i^{\alpha-1}(1-\pi_i)^{\beta-1}/B(\alpha,\beta) d\pi_i \\
&= B(\alpha,\beta)^{-1} \int \pi_i^{(\alpha+2)-1}(1-\pi_i)^{\beta-1} d\pi_i \\
&= B(\alpha+2,\beta) * B(\alpha,\beta)^{-1} \int B(\alpha+1,\beta)^{-1} * \pi_i^{(\alpha+2)-1}(1-\pi_i)^{\beta-1} d\pi_i \\
&= B(\alpha+2,\beta) * B(\alpha,\beta)^{-1} * 1 \\
&= \Gamma(\alpha+2)\Gamma(\beta)/\Gamma(\alpha+2+\beta)/(\Gamma(\alpha)\Gamma(\beta)) * \Gamma(\alpha+\beta) \\
&= \alpha * (\alpha+1)/(\alpha+1+\beta) * (\alpha+\beta) \\
&= \theta(\alpha+1)/(\alpha+1+\beta)
\end{aligned}$$

Then we can obtain $Var(\pi_i)$

$$\begin{aligned}
Var(\pi_i) &= E(\pi_i^2) - E(\pi_i)^2 \\
&= ((\alpha+1)\alpha(\alpha+\beta) - \alpha^2(\alpha+\beta+1))/(\alpha+\beta+1)(\alpha+\beta)^2 \\
&= (\alpha\beta)/(\alpha+\beta)^2/(\alpha+1+\beta) \\
&= \theta(1-\theta)/(\alpha+\beta+1) = \phi\theta(1-\theta)
\end{aligned}$$

c. $E(Y_i) = n_i\theta$

$$\begin{aligned}
E(Y_i) &= E_{\pi_i}(E_{Y_i}(Y_i|\pi_i)) \\
&= E_{\pi_i}(n_i * \pi_i) \\
&= n_i * E(\pi_i) \\
&= n_i * \theta
\end{aligned}$$

d. $Var(Y_i) = n_i\theta(1-\theta)[1+(n_i-1)\phi]$ so that $Var(Y_i)$ is larger than the Binomial variance (unless $n_i = 1$ or $\phi = 0$).

$$\begin{aligned}
Var(Y_i) &= E_{\pi_i}(Var(Y_i|\pi_i)) + Var_{\pi_i}(E(Y_i|\pi_i)) \\
&= E_{\pi_i}(n_i * \pi_i * (1-\pi_i)) + Var_{\pi_i}(\pi_i * n_i) \\
&= n_i * (E(\pi_i) - E(\pi_i^2)) + n_i^2 * \phi\theta(1-\theta) \\
&= n_i * (\theta - \theta(\alpha+1)/(\alpha+1+\beta)) + n_i^2 * \phi\theta(1-\theta) \\
&= n_i * (\theta(1-(\alpha+1)/(\alpha+1+\beta))) + n_i^2 * \phi\theta(1-\theta) \\
&= n_i * (\theta * \beta/(\alpha+1+\beta)) + n_i^2 * \phi\theta(1-\theta) \\
&= n_i * (\theta * (1-\theta)(1-\phi)) + n_i^2 * \phi\theta(1-\theta) \\
&= n_i\theta(1-\theta)[1+(n_i-1)\phi]
\end{aligned}$$

Q2. (ELMR Chapter 3 Exercise 1)

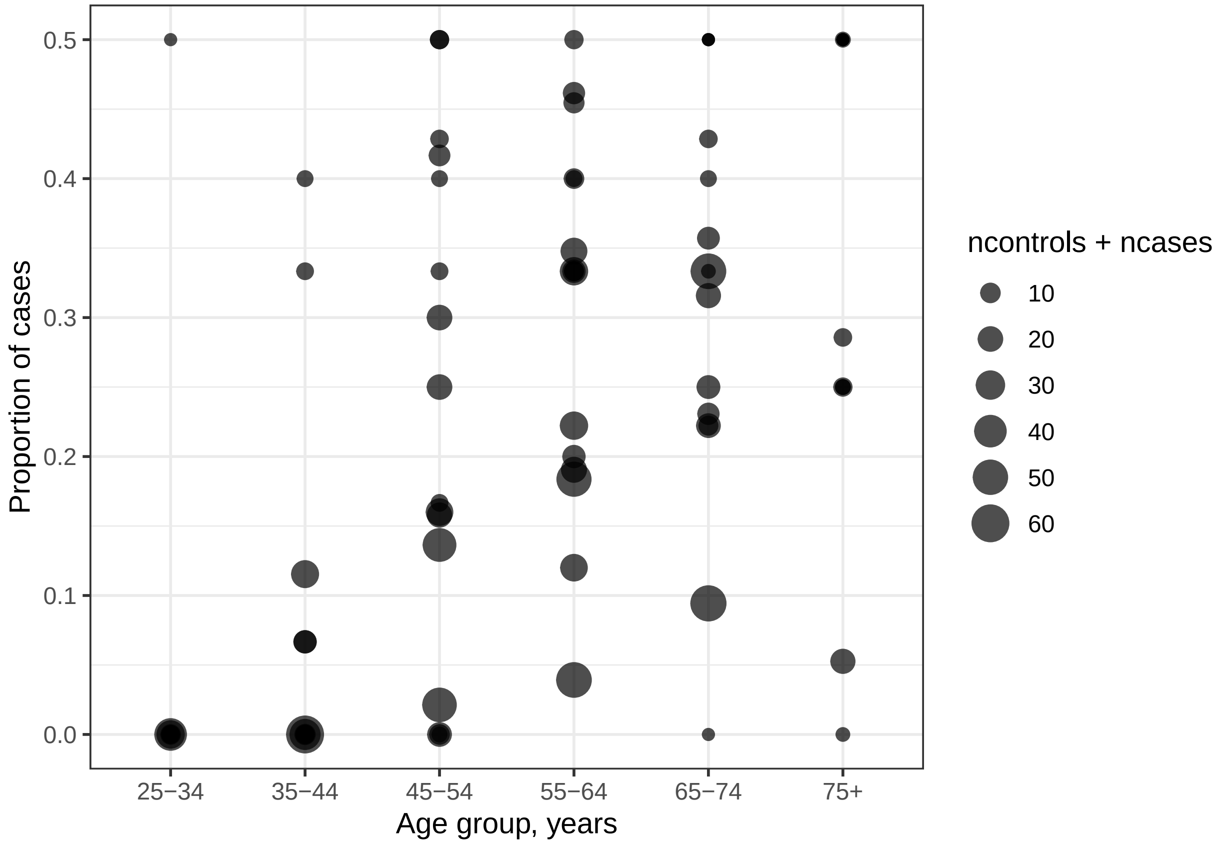
A case-control study of esophageal cancer in Ileet-Vilaine, France.

```
data(esoph)
#help(esoph)
```

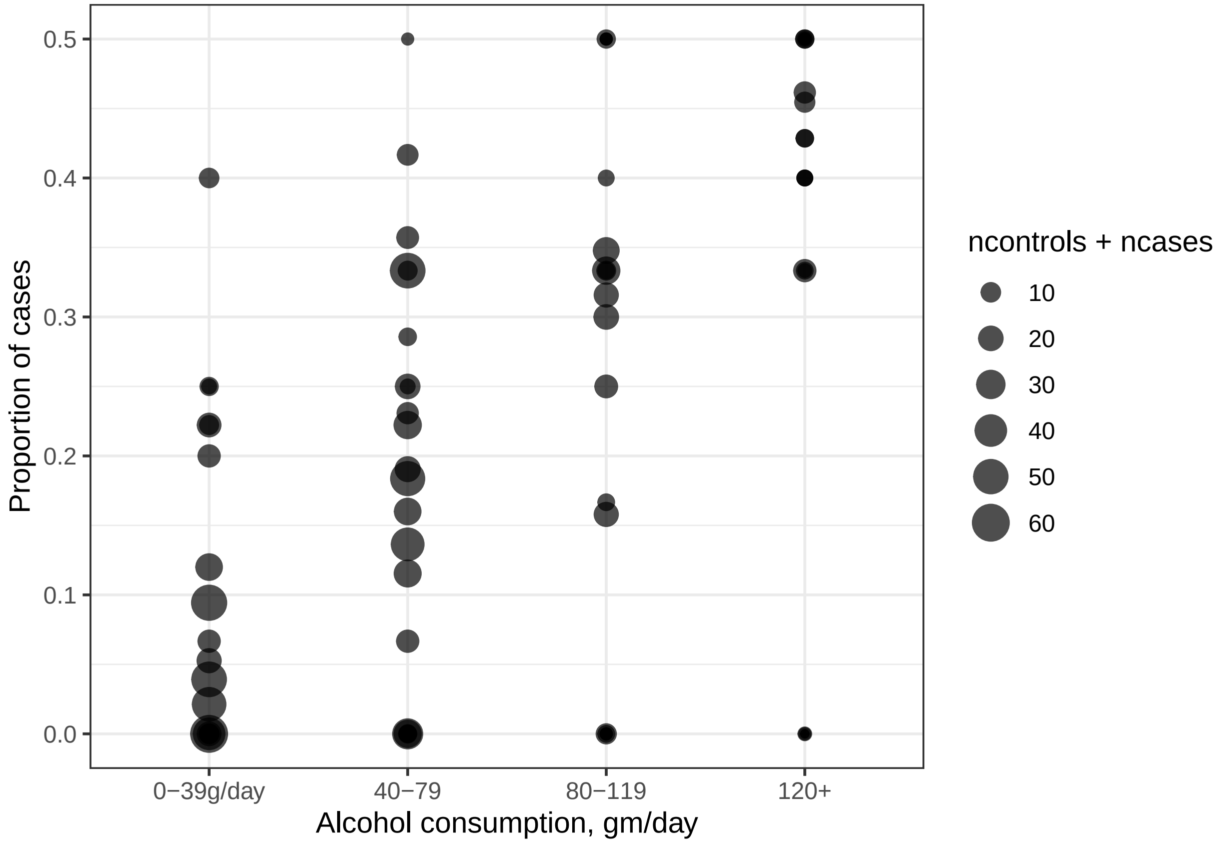
a. Plot the proportion of cases against each predictor using the size of the point to indicate the number of subject as seen in Figure 2.7. Comment on the realtionships seen in the plots.

Solution:

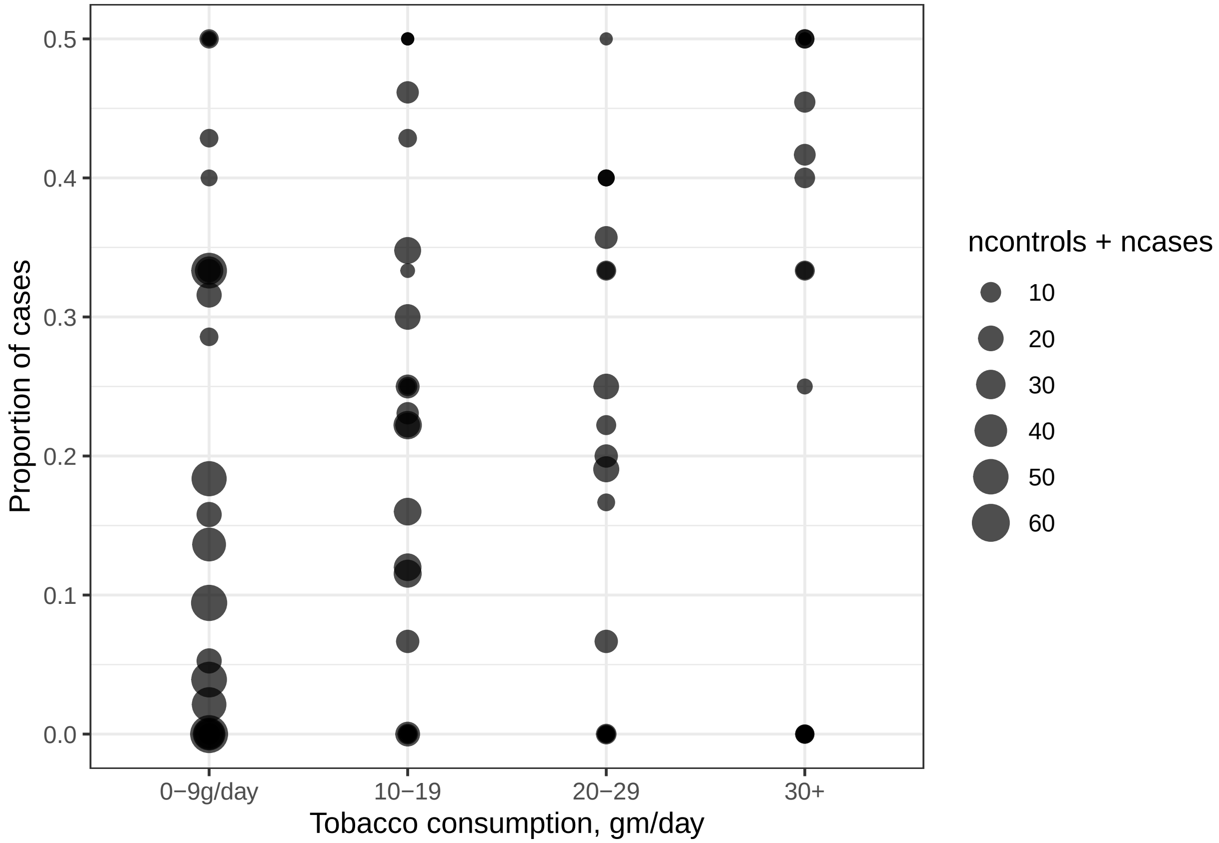
```
plot_data <- esoph %>%
  mutate(proportion=ncases/(ncontrols+ncases))

ggplot(plot_data, aes(agegp, proportion))+
  geom_point(aes(size = ncontrols+ncases),alpha = 7/10)+
  ylab("Proportion of cases")+
  xlab("Age group, years")+
  theme_bw()
```



```
ggplot(plot_data, aes(alcgp, proportion))+
  geom_point(aes(size = ncontrols+ncases),alpha = 7/10)+
  ylab("Proportion of cases")+
  xlab("Alcohol consumption, gm/day")+
  theme_bw()
```



```
ggplot(plot_data, aes(tobgp, proportion))+
  geom_point(aes(size = ncontrols+ncases),alpha = 7/10)+
  ylab("Proportion of cases")+
  xlab("Tobacco consumption, gm/day")+
  theme_bw()
```



b. Fit a binomial GLM with interactions between all three predictors. Use AIC as a criterion to select a model using the `step` function. Which model is selected?

Solution:

```
lmod = glm(cbind(ncases, ncontrols) ~ agegp*alcgp*tobgp,
           family = binomial, data=esoph)
minmod = glm(cbind(ncases, ncontrols) ~ 1,
           family = binomial, data=esoph)
lmod_step = step(lmod, direction = "both")
```

```
## Start:  AIC=323.48
## cbind(ncases, ncontrols) ~ agegp * alcgp * tobgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                       Df Deviance    AIC
## - agegp:alcgp:tobgp 37   16.109 265.59
## <none>                     0.000 323.48

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
```

```
## Step:  AIC=265.59
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp +
##     agegp:tobgp + alcgp:tobgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                     Df Deviance    AIC
## - agegp:tobgp       15   27.146 246.63
## - agegp:alcgp       15   34.364 253.84
## - alcgp:tobgp        9   23.776 255.26
## <none>                   16.109 265.59
## + agegp:alcgp:tobgp 37    0.000 323.48
##
## Step:  AIC=246.63
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp +
##     alcgp:tobgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance    AIC
## - alcgp:tobgp  9   33.796 235.28
## - agegp:alcgp 15   47.484 236.96
## <none>             27.146 246.63
## + agegp:tobgp 15   16.109 265.59
##
## Step:  AIC=235.28
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance    AIC
## - agegp:alcgp 15   53.973 225.45
## <none>             33.796 235.28
## - tobgp        3   44.151 239.63
## + alcgp:tobgp  9   27.146 246.63
## + agegp:tobgp 15   23.776 255.26
##
## Step:  AIC=225.45
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp
##
##               Df Deviance    AIC
## <none>             53.973 225.45
## - tobgp        3   64.572 230.05
## + agegp:alcgp 15   33.796 235.28
## + alcgp:tobgp  9   47.484 236.96
## + agegp:tobgp 15   41.455 242.94
## - alcgp        3  120.028 285.51
## - agegp        5  131.484 292.96
```

```
lmod_step = step(lmod, direction = "backward")
```

```
## Start:  AIC=323.48
## cbind(ncases, ncontrols) ~ agegp * alcgp * tobgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                       Df Deviance    AIC
## - agegp:alcgp:tobgp 37   16.109 265.59
## <none>                     0.000 323.48

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=265.59
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp +
##     agegp:tobgp + alcgp:tobgp

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                 Df Deviance    AIC
## - agegp:tobgp 15   27.146 246.63
## - agegp:alcgp 15   34.364 253.84
## - alcgp:tobgp  9   23.776 255.26
## <none>             16.109 265.59
##
## Step:  AIC=246.63
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp +
##     alcgp:tobgp
##
##                 Df Deviance    AIC
## - alcgp:tobgp  9   33.796 235.28
## - agegp:alcgp 15   47.484 236.96
## <none>             27.146 246.63
##
## Step:  AIC=235.28
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp + agegp:alcgp
##
##                 Df Deviance    AIC
## - agegp:alcgp 15   53.973 225.45
## <none>             33.796 235.28
## - tobgp        3   44.151 239.63
##
## Step:  AIC=225.45
## cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp
##
##         Df Deviance    AIC
## <none>       53.973 225.45
## - tobgp  3   64.572 230.05
## - alcgp  3  120.028 285.51
## - agegp  5  131.484 292.96
```

```
lmod_step = step(minmod, direction = "forward", scope=~agegp*alcgp*tobgp)
```

```
## Start:  AIC=376.72
## cbind(ncases, ncontrols) ~ 1
##
##         Df Deviance    AIC
## + alcgp  3   138.79 294.27
## + agegp  5   139.11 298.59
## + tobgp  3   209.53 365.01
## <none>       227.24 376.72
##
## Step:  AIC=294.27
## cbind(ncases, ncontrols) ~ alcgp
##
##         Df Deviance    AIC
## + agegp  5   64.572 230.05
## + tobgp  3  131.484 292.96
## <none>      138.789 294.27
##
## Step:  AIC=230.05
## cbind(ncases, ncontrols) ~ alcgp + agegp
##
##               Df Deviance    AIC
## + tobgp        3   53.973 225.45
## <none>             64.572 230.05
## + agegp:alcgp 15   44.151 239.63
##
## Step:  AIC=225.45
## cbind(ncases, ncontrols) ~ alcgp + agegp + tobgp
##
##               Df Deviance    AIC
## <none>             53.973 225.45
## + agegp:alcgp 15   33.796 235.28
## + alcgp:tobgp  9   47.484 236.96
## + agegp:tobgp 15   41.455 242.94
```

```
lmod_step %>%
 tbl_regression(intercept = TRUE)
```

Characteristic	**log(OR)**	**95% CI**	**p-value**
(Intercept)	-1.8	-2.3, -1.4	<0.001
alcgp			
alcgp.L	1.5	1.1, 1.9	<0.001
alcgp.Q	-0.23	-0.58, 0.12	0.2
alcgp.C	0.25	-0.06, 0.57	0.11
agegp			
agegp.L	3.0	2.0, 4.8	<0.001
agegp.Q	-1.3	-2.9, -0.39	0.024
agegp.C	0.15	-0.62, 1.3	0.7
agegp^4	0.06	-0.61, 0.66	0.8
agegp^5	-0.19	-0.58, 0.19	0.3
tobgp			

Characteristic	log(OR)	95% CI	p-value
tobgp.L	0.59	0.21, 1.0	0.002
tobgp.Q	0.07	-0.30, 0.43	0.7
tobgp.C	0.16	-0.21, 0.53	0.4

We choose a model by AIC in three Stepwise Algorithms ("both", "backward", "forward"). All of results provides the same best model. Thus, we selected `cbind(ncases, ncontrols) ~ agegp + alcgp + tobgp` as the best model.

c. All three factors are ordered and so special contrasts have been used appropriate for ordered factors involving linear, quadratic and cubic terms. Further simplification of the model may be possible by eliminating some of these terms. Use the `unclass` function to convert the factors to a numerical representation and check whether the model may be simplified.

Solution:

```
lmod = glm(cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp)
           + unclass(tobgp), family = binomial, data=esoph)
lmod_unclass = step(lmod, direction = "both")
```

```
## Start:  AIC=229.44
## cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp) +
##     unclass(tobgp)
##
##                  Df Deviance    AIC
## <none>                73.959 229.44
## - unclass(tobgp)  1   85.310 238.79
## - unclass(agegp)  1  135.311 288.79
## - unclass(alcgp)  1  146.355 299.84
```

```
lmod_unclass = step(lmod, direction = "backward")
```

```
## Start:  AIC=229.44
## cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp) +
##     unclass(tobgp)
##
##                  Df Deviance    AIC
## <none>                73.959 229.44
## - unclass(tobgp)  1   85.310 238.79
## - unclass(agegp)  1  135.311 288.79
## - unclass(alcgp)  1  146.355 299.84
```

```
lmod_unclass = step(minmod, direction = "forward",
             scope=~unclass(agegp)+unclass(alcgp)+unclass(tobgp))
```

```
## Start:  AIC=376.72
## cbind(ncases, ncontrols) ~ 1
##
##                  Df Deviance    AIC
## + unclass(alcgp)  1   142.21 293.69
## + unclass(agegp)  1   167.59 319.07
## + unclass(tobgp)  1   211.22 362.70
## <none>                227.24 376.72
##
## Step:  AIC=293.69
## cbind(ncases, ncontrols) ~ unclass(alcgp)
##
##                  Df Deviance    AIC
## + unclass(agegp)  1    85.31 238.79
## + unclass(tobgp)  1   135.31 288.79
## <none>                142.21 293.69
##
## Step:  AIC=238.79
## cbind(ncases, ncontrols) ~ unclass(alcgp) + unclass(agegp)
##
##                  Df Deviance    AIC
## + unclass(tobgp)  1   73.959 229.44
## <none>                85.310 238.79
##
## Step:  AIC=229.44
## cbind(ncases, ncontrols) ~ unclass(alcgp) + unclass(agegp) +
##     unclass(tobgp)
```

```
lmod_unclass %>%
 tbl_regression(intercept = TRUE)
```

Characteristic	**log(OR)**	**95% CI**	**p-value**
(Intercept)	-5.6	-6.4, -4.8	<0.001
unclass(alcgp)	0.69	0.53, 0.86	<0.001
unclass(agegp)	0.53	0.39, 0.67	<0.001
unclass(tobgp)	0.27	0.12, 0.43	<0.001

We choose a model by AIC in three Stepwise Algorithms ("both", "backward", "forward"). All of results provides the same best model. Thus, we selected `cbind(ncases, ncontrols) ~ unclass(agegp) + unclass(alcgp) + unclass(tobgp)` as the best model.

d. Use the summary output of the factor model to suggest a model that is slightly more complex than the linear model proposed in the previous question.

Solution:

```
#refer to original factor model
lmod = glm(cbind(ncases, ncontrols) ~ agegp*alcgp*tobgp,
           family = binomial, data=esoph)
#final model
lmod_final = glm(cbind(ncases, ncontrols) ~ unclass(alcgp) +
                   agegp + unclass(tobgp), family = binomial, data=esoph)
lmod_final %>%
 tbl_regression(intercept = TRUE)
```

Characteristic	**log(OR)**	**95% CI**	**p-value**
(Intercept)	-4.0	-4.7, -3.4	<0.001
unclass(alcgp)	0.65	0.49, 0.82	<0.001
agegp			
agegp.L	3.0	1.9, 4.7	<0.001
agegp.Q	-1.3	-2.9, -0.40	0.023
agegp.C	0.15	-0.62, 1.3	0.7
agegp^4	0.07	-0.61, 0.66	0.8
agegp^5	-0.20	-0.59, 0.18	0.3
unclass(tobgp)	0.26	0.10, 0.42	0.001

According to the summary output of the factor model, we found that quadratic term of age group is significant within the 95% confidence interval. In addition, tobgp and alcgp only have significant linear terms. Thus, we kept agegp as ordered categorical variable and unclassed alcgp and tobgp.

e. Does your final model fit the data? Is the test you make accurate for this data?

Solution:

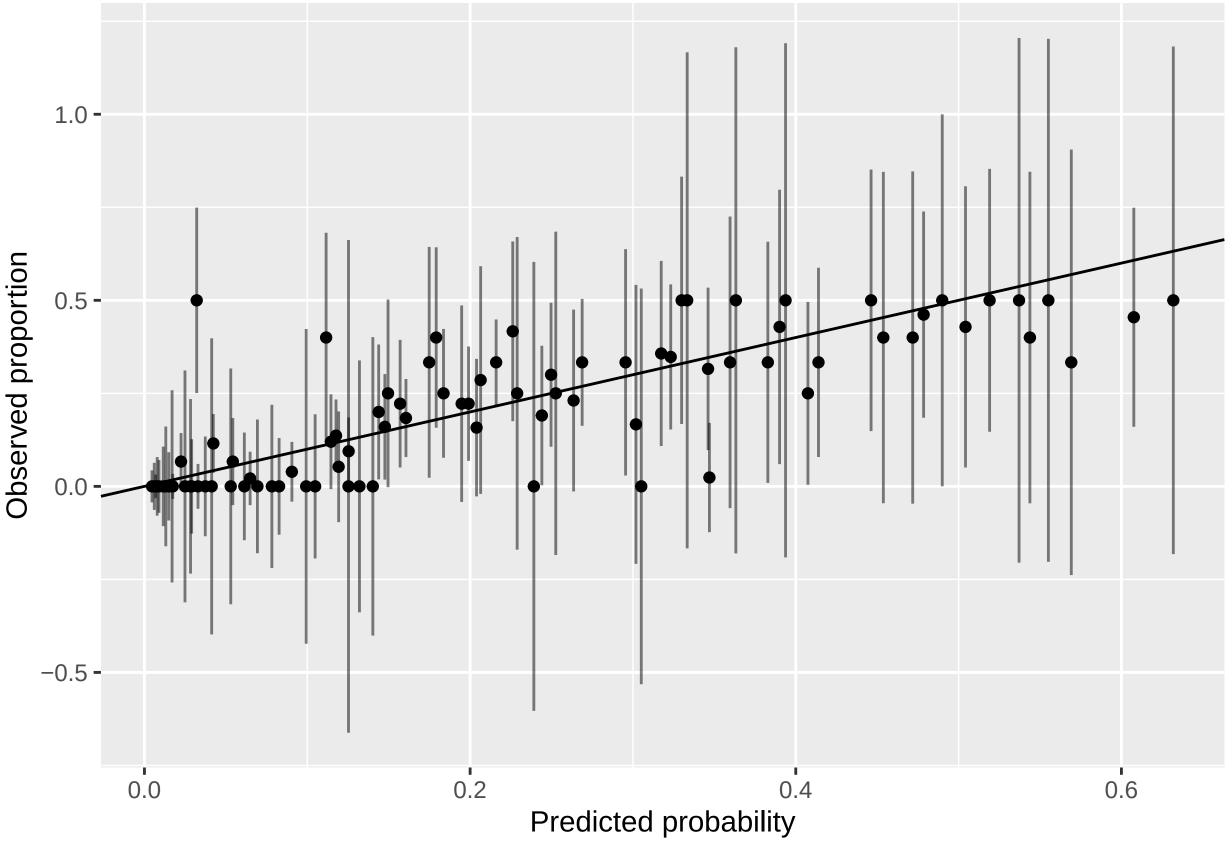
```
# test the deviance
pchisq(lmod_final$deviance, lmod_final$df.residual, lower = FALSE)
```

```
## [1] 0.9601137
```

```
df <- esoph %>%
  mutate(proportion=ncases/(ncontrols+ncases)) %>%
  mutate(weight=(ncontrols+ncases))
predprob <- predict(lmod_final, type = "response")

# Pearson chi-square statistic
px2 <- sum((df$ncases - df$weight*predprob)^2 /
             (df$weight*predprob*(1 - predprob)))
pchisq(px2, lmod_final$df.residual, lower.tail = FALSE)
```

```
## [1] 0.9271845
```

```
# Hosmer-Lemeshow test statistic
df_binned <- df %>%
  mutate(predprob = predict(lmod_final, type = "response"),
         linpred = predict(lmod_final, type = "link"),
         bin = cut(linpred,
                   breaks = unique(quantile(linpred, (1:100) / 101)))) %>%
  group_by(bin) %>%
  summarize(y = sum(ncases),
            avgpred = mean(predprob),
            count = sum(weight)) %>%
  mutate(se_fit = sqrt(avgpred * (1 - avgpred) / count))
df_binned %>%
  ggplot(mapping = aes(x = avgpred, y = y / count)) +
  geom_point() +
  geom_linerange(mapping = aes(ymin = y / count - 2 * se_fit,
                               ymax = y / count + 2 * se_fit), alpha = 0.5) +
  geom_abline(intercept = 0, slope = 1) +
  labs(x = "Predicted probability", y = "Observed proportion")
```



```
# Hosmer-Lemeshow test
hlstat <- with(df_binned, sum((y - count * avgpred)^2 /
                              (count * avgpred * (1 - avgpred))))
# J
nrow(df_binned)
```

```
## [1] 87
```

```
# p-value
pchisq(hlstat, nrow(df_binned) - 1, lower.tail = FALSE)
```

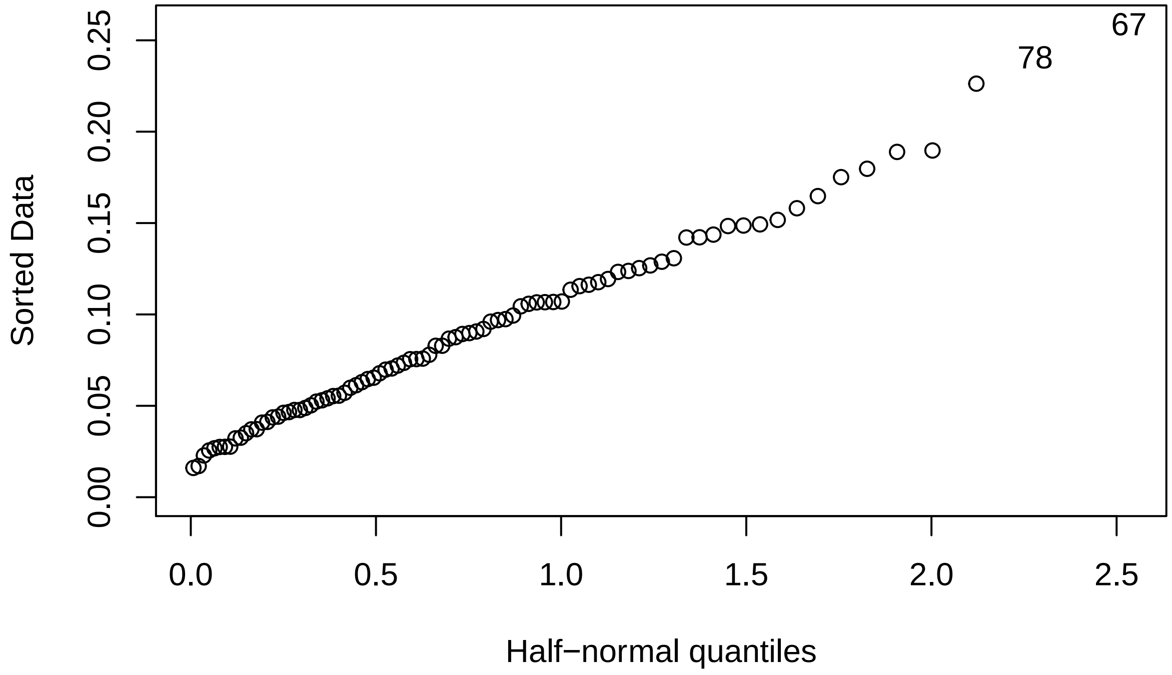
```
## [1] 0.6239636
```

We conducted pearson chi-squre test on the deviance D, Pearson chi-square statistic and Hosmer-Lemeshow test statistic. All of them present large p-value, indicating that the model has an adequate fit.

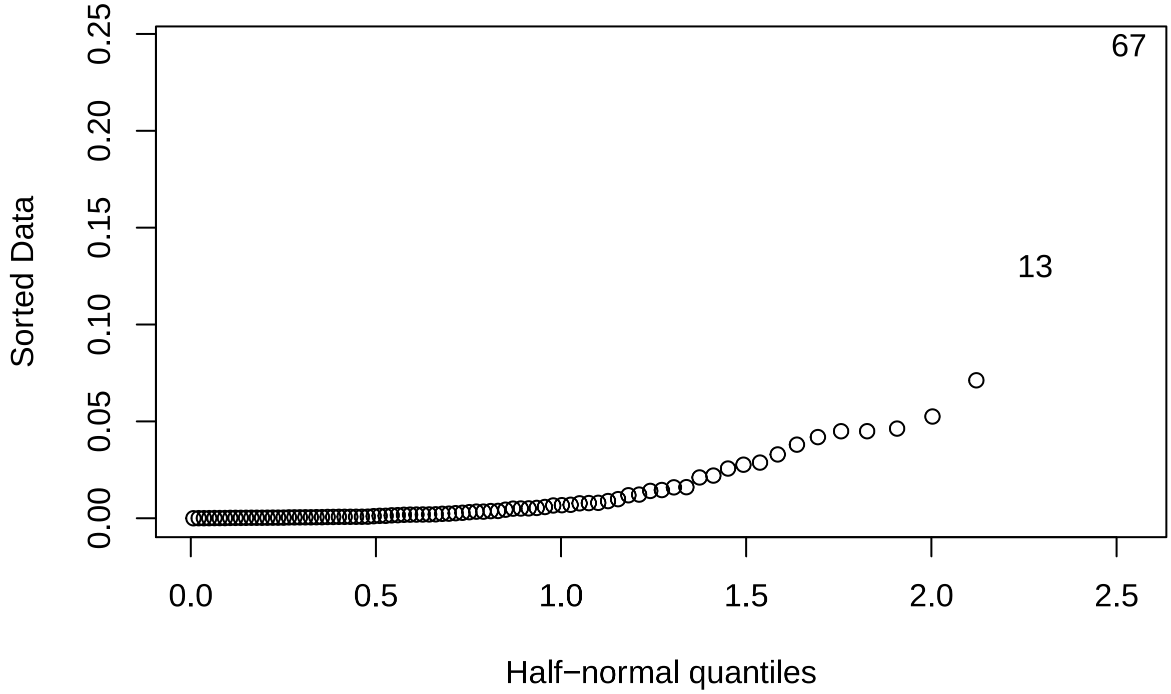
f. Check for outliers in your final model.

Solution:

```
df %>%
  mutate(devres = residuals(lmod_final, type = "deviance"))%>%
  mutate(linpred = predict(lmod_final, type = "link")) -> df
halfnorm(hatvalues(lmod_final))
```



```
halfnorm(cooks.distance(lmod_final))
```



According to the hatvalues and cooks.distance plots, we identified potential high influential observations (13, 67, 78).

Then we print out outliers and check:

```
df %>%
  slice(c(13, 67, 78))
```

```
##   agegp     alcgp    tobgp ncases ncontrols proportion weight     devres
## 1 25-34      120+    10-19      1         1 0.50000000      2  2.0421345
## 2 65-74     40-79 0-9g/day     17        34 0.33333333     51  1.9307622
## 3   75+ 0-39g/day 0-9g/day      1        18 0.05263158     19 -0.9946984
##     linpred
## 1 -3.406193
## 2 -1.289175
## 3 -1.999424
```

g. What is the predicted effect of moving one category higher in alcohol consumption?

Solution:

```
coefs <- coef(lmod_final)
odds = exp(coefs[2] * 1)
round(as.numeric(odds),2)
```

```
## [1] 1.92
```

According to the results, we know that the risk of moving one category higher in alcohol consumption, the risk would be 92% higher.

h. Compute a 95% confidence interval for this predicted effect.

Solution:

```
confint(lmod_final)
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %     97.5 %
## (Intercept)    -4.6842003 -3.4325049
## unclass(alcgp)  0.4888562  0.8205436
## agegp.L         1.9110876  4.7249613
## agegp.Q        -2.9427677 -0.3971914
## agegp.C        -0.6234592  1.2959116
## agegp^4        -0.6105855  0.6594479
## agegp^5        -0.5890733  0.1820650
## unclass(tobgp)  0.1003185  0.4220752
```

```
odds_lower = exp(0.4888562 * 1)
odds_upper = exp(0.8205436 * 1)

#95% confidence interval
round(as.numeric(c(odds_upper, odds_lower)), 2)
```

```
## [1] 2.27 1.63
```